intel

ARTICLE REPRINT

AR-189

November 1981

Keep memory design simple yet cull single-bit errors

M. Bazes
L. Farrell
B. May
M. Mebel
Intel Corporation

AFN-02124A
ORDER NUMBER: 210287-001

A new error-correction chip with dual-bus architecture interfaces easily with dynamic RAMs. Memory-system reliability soars and the additional parts count is relatively modest.

Keep memory design simple yet cull single-bit errors

In memory-system design, the demand for greater reliability is reflected by an increasing interest in error-detection and correction circuitry. Several semiconductor manufacturers have recently introduced error-detection and correction chips. They share a common architecture that features a multiplexed data bus. But the Intel 8206 error-detection and correction unit (EDCU) is different: This LSI device, fabricated in HMOS II, allows error correction to be added to memory systems with minimal overhead.

A single 8206 handles 8 or 16-bit data widths, and up to five 8206's can be cascaded to handle all multiples of eight bits (up to 80 bits). The 8206 corrects single-bit errors in a maximum of 65 ns for 16-bit systems and typically replaces 20 to 40 ICs, depending upon the number of features in the system.

Common error detection circuits simply recognize that data has a parity error. Correction circuits use the Hamming code as an extension of parity to detect and give the position of the error, allowing it to be corrected.

Single-bit correction and multiple-bit detection is the typical implementation, reflecting the tradeoff between the probability of errors in a system and the cost of additional memory. For a 16-bit system, single-bit error correction and double-bit error detection is implemented by using 6 additional check bits, for an overhead of 37% (Table 1). Adding single-bit error correction to a system improves system reliability by at least a factor of 24 (Table 2).

M. Bazes, Design Engineer
L. Farrell, Marketing Manager
B. May, Applications Engineer
M. Mebel, Design Engineer
Intel Corp., 3065 Bowers Ave.
Santa Clara, CA 95051

Error correction is used extensively in mainframe and minicomputer design where memory sizes of several megabytes are common. Here the probability of error is directly related to the error rate of the individual RAMs and the number of RAMs in the system. As the number of RAMs increases, so does the system error rate.

With today's microprocessors, like the Intel eight-bit iAPX 88 and 16-bit iAPX 86 (each can directly address 1 Mbyte), typical RAM memory sizes are 100 kbytes and climbing. As a result, microprocessor system designers are looking to add error correction as simply as possible.

New bus architecture

The 8206 is the first 16-bit EDCU to use separate input and output data buses, a feature that simplifies system design, saves board space, and reduces parts count. The new architecture is made possible by packaging the 8206 in a JEDEC type A 68-pin leadless chip carrier. Figure 1 shows the 8206's functional blocks.

During read cycles, data and check bits enter via the data input (DI) and check-bit input (CBI) pins, where they are optionally latched by the STB input. The data then take two parallel paths. The first path is to the data-output (DO/WDI) pins, where the uncorrected data are available 32 ns later. The second path is to the check-

September 30, 1981

bit generator, where check bits generated from the data are compared with the check bits read from the memory.

The result of the comparison is the *syndrome*, a 5-to-8-bit value identifying which bit (if any) was in error. The syndrome is then decoded to a 1-of-16 bit strobe which is used to "flip" the bit in error (assuming the $\overline{CRCT}$ input is active). Syndrome decoding also tells the 8206 whether to assert the error flags. The 16 data output pins are enabled on a byte basis by the $\overline{BM}$ inputs.

For write cycles, data enter the write data input (DO/WDI) pins and goes to the check-bit generator. The check bits are then written to the check-bit memory by the check-bit output (SYO/CBO/PPO) pins. These pins also output the syndrome bits during read or read-modify-write cycles.

Note that only the 8206's R/$\overline{W}$ pin is typically used for control during a memory cycle. This pin informs the 8206 whether the cycle is a read (generate new check bits and compare to those from memory) or a write (generate new check bits only). During a read-modify-write cycle, a falling edge of R/$\overline{W}$ tells the 8206 to latch the syndrome bits internally and output check bits to be written back into memory. The strobe input (STB) may optionally be used to latch data and check bits internally.

The 8206's dual-bus architecture saves the additional control lines and the sequencing logic required

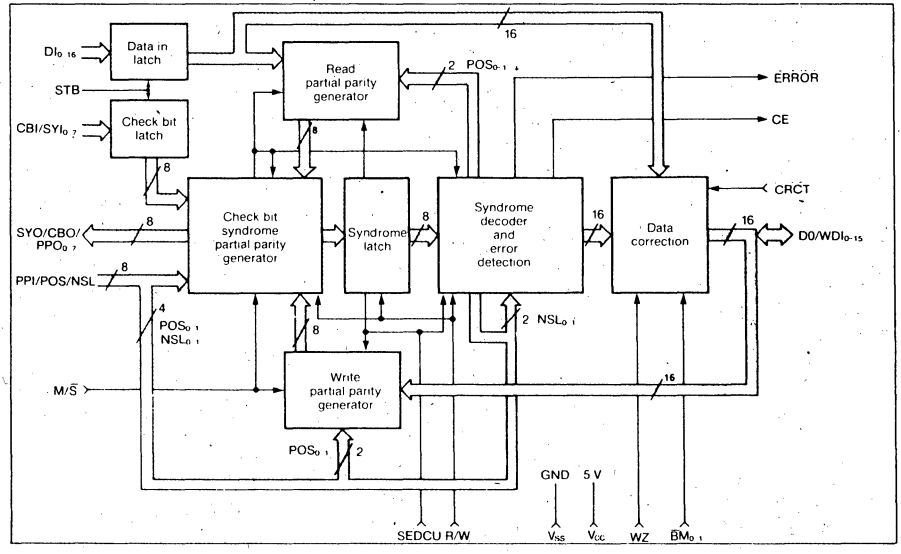
Table 1. Check bits required for single-bit correction, multiple-bit detection.

Data word bits	Check bits	Overhead % (# check bits/# data bits)
8	5	62
16	6	37
32	7	22
64	8	12
80	8	10

Table 2. Single-bit error correction increases memory reliability a minimum of 24 times.

*Memory size	MTBF (no error correction)	MTBF (single-bit error correction)	MTBF improvement ratios
32 kbytes	5.6 Years	133.6 Years	24
64 "	2.7 "	75.1 "	28
128 "	1.4 "	40.5 "	29
5 Mbytes	16 Days	10.8 "	246
8 "	8 "	6.1 "	278
16 "	4 "	3.3 "	301

*Based on a 16 kbit dynamic RAM with a failure rate of 0.127% every 1000 hours.
Note: MTBF, though related to memory size, also depends on memory organization (e.g. word width, number of pages) that is not detailed in this table.



1. The 8206's two 16-bit data buses, one for data from the RAM (DI_{0-15}) and one for data to the system bus (DO_{0-15}), minimize the external control logic required.

by single-bus EDCUs. The principal advantages of dual-bus architecture can be illustrated by looking at the three types of memory cycles: reads, writes, and read-modify-writes.

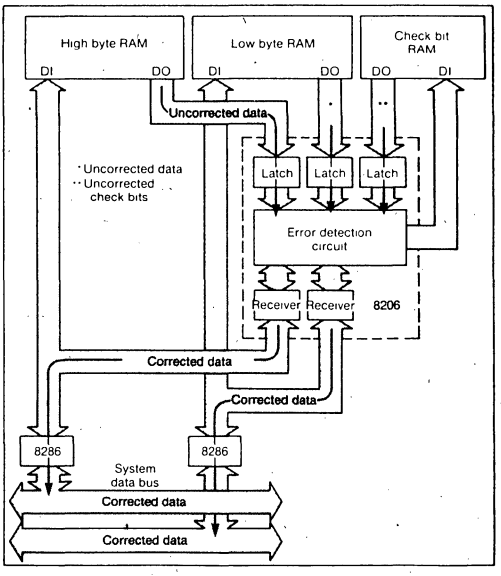
In a read cycle (Fig. 2), data and check bits are received from the RAM outputs by the DI and CBI pins. New check bits are generated from the data bits and compared to the check bits read from the RAM. An error in either the data or the check bits read from memory means the generated check bits will not match the read check bits. If an error is detected, the $\overline{\text{ERROR}}$ flag is activated and the correctable error (CE) flag tells the system if the error is (or is not) correctable.

With the $\overline{\text{BM}}$ inputs high, the corrected word appears at the DO pins (if the error was correctable), or the unmodified word appears (if the error was uncorrectable). Note that for this correction cycle there is no control or timing logic required. The 8206's dual buses isolate the RAM outputs from the EDCU outputs. Special transceivers that prevent contention between the uncorrected RAM data and corrected EDCU data are not needed.

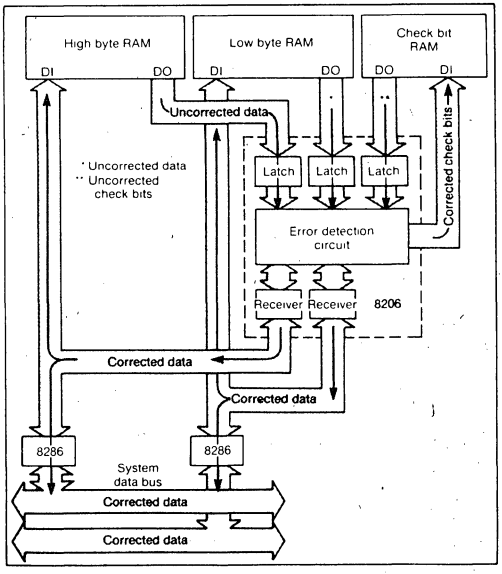
A syndrome word, five to eight bits in length and containing all necessary information about the existence and location of an error, is provided at the syndrome output (SYO/CBO/PPO) pins. Error logging is accomplished by latching the syndrome and the memory address of the word in error. The syndrome decoding of Table 3 can be used as a table lookup by the CPU.

If an error is detected during a read, the read cycle is extended to a read-modify-write cycle where the corrected data is rewritten to the same location. This offers several advantages:

- Since soft errors are random, independent processes, the longer a soft error is allowed to remain in memory, the greater the probability that a second soft error will occur in the memory word, resulting in an uncorrectable double-bit error. By writing the correct data back to RAM, the mean "lifetime" of soft errors is reduced, greatly reducing the chance of double-bit errors, and increasing reliability.
- "Error scrubbing" (going through the entire memory and correcting any soft errors) may be done as a background software task. For instance, the 8086 microprocessor's load string (LODS) instruction can consecutively read all addresses in RAM. Any soft errors will be corrected. Scrubbing further increases system reliability.
- Error logging may be used to detect hard errors. (A soft error is seen once when the affected word is read and is then corrected, while a hard error is seen again and again.) An error logger shows a consistent pattern if a hard error is present in a particular word. A system may be configured to generate an interrupt when the 8206 detects an error.



2. The 8206 requires no control logic or timing inputs to perform read-with-correction cycles.



3. The 8206 can correct both data bits and check bits.

This last advantage allows the operating system to re-read the address where the error occurred. If the same error re-occurs, it is assumed to be a hard error, and while the system can continue to function, maintenance is indicated. The operating system may mark that page of memory as "bad" until its PC card

has been serviced. Alternatively, the memory system may reconfigure itself and map the bit where the hard error occurred to a spare dynamic RAM whenever the affected memory page is accessed.

When a correctable error occurs during a read cycle (Fig. 3), the system's dynamic RAM controller (or CPU) examines the 8206 ERROR and CE outputs to determine if a correctable error occurred. If it did, the dynamic RAM controller (or CPU) forces R/$\overline{W}$ low, telling the 8206 to latch the generated syndrome and drive the corrected check bits onto the SYO/CBO/PPO outputs. The corrected data is already available on the DO/WDI pins. The dynamic RAM controller then writes the corrected data and check bits into memory. Once again the 8206's dual buses allow this cycle to be implemented without special bus transceivers.

The 8206 may be used to perform read-modify-writes in one or two RAM cycles. If it is done in two cycles, the 8206 latches are used to hold the data and check bits from the read cycle to be used in the immediately following write cycle.

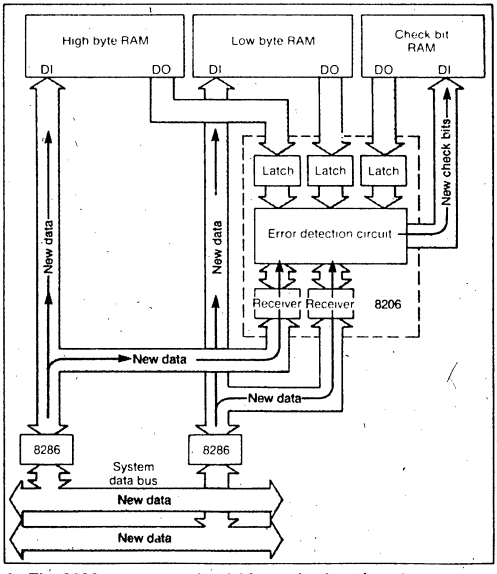
Write cycle corrections

For a full-word write (Fig. 4) where an entire word is written to memory, data are written directly to the RAM. This same data enter the 8206 through the DO/WDI pins where five to eight check bits are generated. The check bits are then sent to the RAM through the SYO/CBO/PPO pins for storage along with the data word.

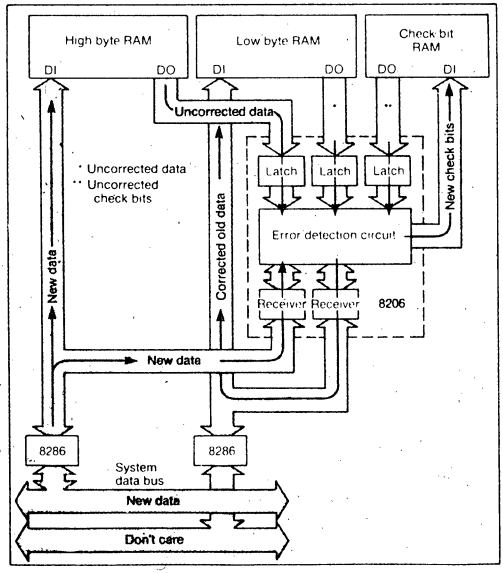
A byte write (Fig. 5) is implemented as a read-modify-write cycle. Since the Hamming code works only on entire words, to write one byte of the word, it is necessary to read the entire word to be modified, perform error correction, merge the new byte into the old word inside the 8206, generate check bits for the new word, and write the whole word plus check bits into RAM.

Error correction on the old word is important. Suppose a bit error occurs in the half of the old word that was not changed. This old byte would be combined with the new byte, and check bits would be generated for the whole word, including the bit in error. The bit error now becomes "legitimate"; no error will be detected when this word is read, and the system may crash. Obviously, it is important to eliminate this bit error before new check bits are generated.

The 8206 may alternatively be used in a "check-only" mode with the correct ($\overline{CRCT}$) pin left inactive. With the correction facility turned off, the delay of generating and decoding the syndromes is avoided, and the propagation delay from memory outputs to 8206 outputs is significantly shortened. In the event of an error, the 8206 activates the $\overline{ERROR}$ flag to the CPU or dynamic RAM controller, which can then perform one of several optins: lengthen the current cycle for correction, restart the instruction, perform a diagnostic routine, or activate the $\overline{CRCT}$ input to enable error correction. Even with the $\overline{CRCT}$ pin



4. The 8206 generates check bits and writes them to memory.



5. The "new data" byte is supplied by the CPU, while the 8206 supplies the corrected old byte. The 8206 also generates new check bits.

inactive, the 8206 generates and decodes the syndrome bits, so that data may be corrected rapidly if the $\overline{CRCT}$ is activated.

Multiple 8206 systems

A single 8206 handles eight or 16 bits of data and five or six check bits, respectively. Up to five 8206's can be cascaded for 80-bit data words with eight check bits. When cascaded, one 8206 operates as a master, and all others work as slaves (Fig. 6).

As an example, during a read cycle in a 32-bit system with one master and one slave, the slave calculates "partial parity" on its portion of the word and presents it to the master through the partial-parity output (SYO/CBO/PPO) pins. The master receives the partial parity at its partial-parity input (PPI/POS/NSL) pins and combines the partial parity from the slave with the parity it calculated from its own portion of the word to generate the syndrome. The syndrome is then returned from the master to the slave for error correction.

The 8206 uses a modified Hamming code which was optimized for multi-chip EDCU systems. The code is such that partial parity is computed by all 8206's in parallel. No 8206 requires more time for logic propagation than any other, hence no single device becomes a bottleneck in the parity operation.

The 8206 is easy to use with all kinds of dynamic RAM controllers. Because of its dual-bus architecture, the amount of control logic needed is very small.

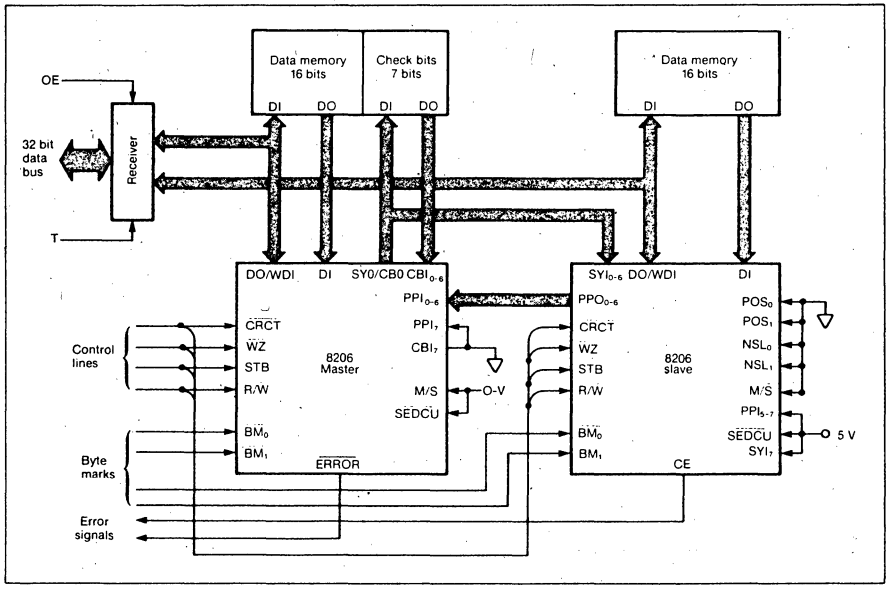
Figure 7a shows a memory design using the 8206 with Intel's 8203 64-kbit dynamic RAM controller and 2164 64-kbit dynamic RAM. As few as three additional ICs complete the memory control function (Fig. 7b).

For simplicity, all memory cycles are implemented as single-cycle read-modify-writes (Fig. 8). This cycle differs from a normal read or write primarily in when the RAM Write Enable ($\overline{WE}$) is activated. In a normal write cycle, $\overline{WE}$ is activated early in the cycle. In a read cycle, $\overline{WE}$ is inactive.

A read-modify-write cycle consists of two phases. In the first phase, $\overline{WE}$ is inactive, and data are read from the RAM; for the second phase, $\overline{WE}$ is activated and the (modified) data is written into the same word in the RAM. Dynamic RAMs have separate data input and output pins so that modified data may be written, even as the original data is being read. Therefore, data may be read and written in only one memory cycle.

In order to perform read-modify-writes in one cycle, the 8203 dynamic RAM's CAS strobe must be active long enough for the 8206 to access and correct data from the RAM, and write the corrected data back into RAM. CAS active time (t_{CAS}) depends on the 8203's clock frequency. The clock frequency and dynamic RAM must be chosen to satisfy:

$$t_{CAS(min)}^{8203} \geq t_{CAC}^{RAM} + TDVQV^{8206} + TQVQV^{8206} + t_{DS}^{RAM} + t_{CWL}^{RAM}$$



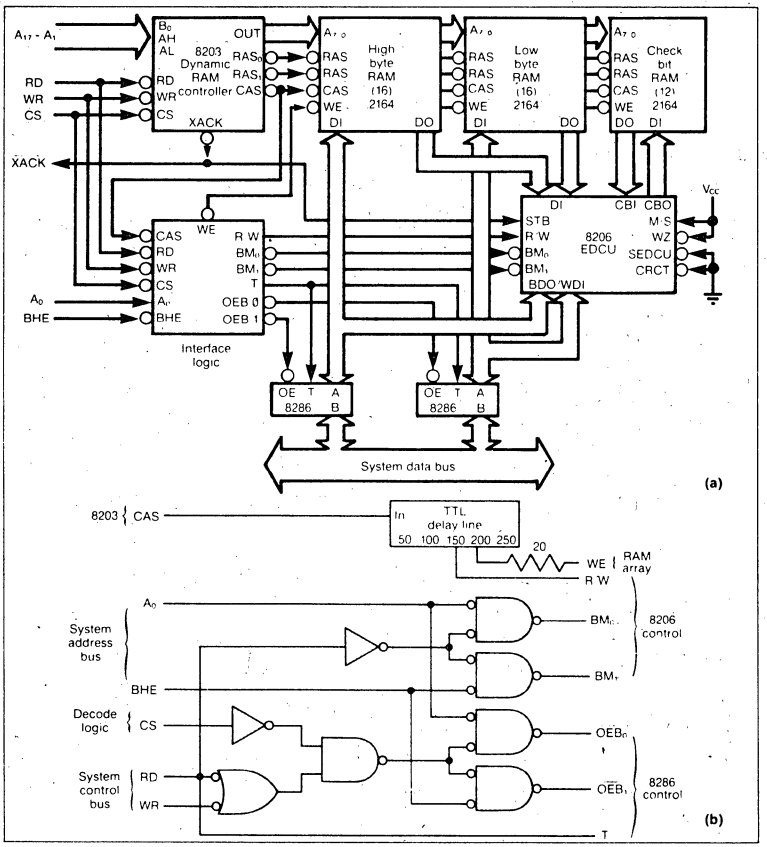
6. No additional logic is required for this 32-bit master-slave system. The slave calculates partial parity on its half of the data, and the master determines which of the 32 data bits and 7 check bits is in error.

Table 3. Syndrome decoding identifies and corrects all single-bit errors.

Syndrome bits 7	6	5	4	0 0 1 0 2 0 3 0	1 0 0 0 0 0	0 1 0 0 0 0	1 1 0 0 0 0	0 0 1 1 0 0	1 0 1 0 0 0	0 1 1 1 0 0	1 1 0 0 0 0	0 0 1 1 0 1	1 0 1 1 0 1	0 1 1 1 0 1	1 1 0 0 1 1	0 0 1 1 1 1	1 0 1 1 1 1	0 1 1 1 1 1	1 1 1 1 1 1
0	0	0	0	N	CB0	CB1	D	CB2	D	D	18	CB3	D	D	0	D	1	2	D
0	0	0	1	CB4	D	D	5	D	6	7	D	D	3	16	D	4	D	D	17
0	0	1	0	CB5	D	D	11	D	19	12	D	D	8	9	D	10	D	D	67
0	0	1	1	D	13	14	D	15	D	D	21	20	D	D	66	D	22	23	D
0	1	0	0	CB6	D	D	25	D	26	49	D	D	48	24	D	27	D	D	50
0	1	0	1	D	52	55	D	51	D	D	70	28	D	D	65	D	53	54	D
0	1	1	0	D	29	31	D	64	D	D	69	68	D	D	32	D	33	34	D
0	1	1	1	30	D	D	37	D	38	39	D	D	35	71	D	36	D	D	U
1	0	0	0	CB7	D	D	43	D	77	44	D	D	40	41	D	42	D	D	U
1	0	0	1	D	45	46	D	47	D	D	74	72	D	D	U	D	73	U	D
1	0	1	0	D	59	75	D	79	D	D	58	60	D	D	56	D	U	57	D
1	0	1	1	63	D	D	62	D	U	U	D	D	U	U	D	61	D	D	U
1	1	0	0	D	U	U	D	U	D	D	U	76	D	D	U	D	U	U	D
1	1	0	1	78	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	0	U	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	1	D	U	U	D	U	D	D	U	U	D	D	U	D	U	U	D

N = No error
CBX = Error in check bit X (correctable)
X = Error in data bit X (correctable)

D = Double-bit error (detected but not corrected)
U = uncorrectable multi-bit error



7. The 256-kbyte system (a) has 32 64-kbyte dynamic RAMs for data plus 12 dynamic RAMs for error correction. The dynamic RAMs are controlled by the 8203 dynamic RAM controller while error correction control is supplied by the 8206. Interface logic (b) allows the 8203/8206 system to implement read-modify-write cycles by generating Write Enable ($\overline{WE}$) to the RAMs, Read/Write ($R/\overline{W}$) to the 8206, and byte-control signals.

The 8203 itself performs normal reads and writes. To perform read-modify-writes, simply change the timing of the $\overline{WE}$ signal. In Fig. 7b, $\overline{WE}$ is generated by the interface logic—the 8203 $\overline{WE}$ output is not used. All other dynamic RAM control signals come from the 8203. A 20-Ω damping resistor reduces the $\overline{WE}$ signal ringing. These damping resistors are included on-chip for all 8203 outputs.

The interface logic generates the R/$\overline{W}$ input tothe 8206. This signal is high for read cycles and low for write cycles. During a read-modify-write cycle, R/$\overline{W}$ is first high, then low.

The falling edge of R/$\overline{W}$ tells the 8206 to latch its syndrome bits internally and generate corrected check bits to be written to RAM. Corrected data are already available from the DO pins. No control signals at all are required to generate corrected data. R/$\overline{W}$ is generated by delaying CAS from the 8203 with TTL-buffered delay line. This delay ($t_{DELAY\ 1}$) must satisfy:

$$t_{DELAY\ 1} \geq t_{CAC}{}^{RAM} + TDVRL^{8206}$$

The 8206 uses multiplexed pins to output the syndrome word and then the check bits. The R/$\overline{W}$ signal may be used to latch the syndrome word externally for error logging. The 8206 also supplies two useful error signals: $\overline{ERROR}$ indicates an error is present in the data or check bits; CE tells if the error is correctable (single bit) or uncorrectable (multiple bits).

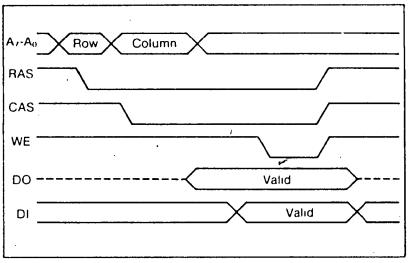
After R/$\overline{W}$ goes low, sufficient time is allowed for the 8206 to generate corrected check bits, then the interface logic activates $\overline{WE}$ to write both corrected data and check bits into RAM. $\overline{WE}$ is generated by delaying $\overline{CAS}$ from the 8203 with the same delay line used to generate R/$\overline{W}$. This delay, $t_{DELAY\ 2}$, must be long enough to allow the 8206 to generate valid check bits, but not so long that the spec of the RAM (t_{CWL}) is violated. This is expressed by:

$$t_{DELAY\ 1} + TRVSV^{8206} \leq t_{DELAY\ 2} \leq t_{CAS(min)}^{8203} - t_{CWL}{}^{RAM}$$

Errors in both data and check bits are automatically corrected, without special 8206 programming.

Since the 8203 terminates $\overline{CAS}$ to the RAMs at a fixed interval after the start of a memory cycle, a latch is usually needed to maintain data on the bus until the 8086 completes the read cycle. This is conveniently done by connecting XACK from the 8203 to the STB input of the 8206, latching the read data and check bits inside the 8206.

The 8086, like all 16-bit CPUs, is capable of reading and writing single-byte data to memory. As just explained, the Hamming code works only on entire words, so in byte writes, and new byte and old byte must be merged, and new check bits written for the composite word. This is difficult with most EDC chips, but it is easy with the 8206.



8. In all memory cycles, the row and column addresses are strobed to the RAMs by $\overline{RAS}$ and $\overline{CAS}$. Sometime after the data out is valid, the control logic in Fig. 7b generates Write Enable ($\overline{WE}$) to write the data back into the RAMs.

Further qualifications on 8206 operation

Referring again to Fig. 7b, the 8206 byte-mark inputs ($\overline{BM_0}$, $\overline{BM_1}$), are generated from A0 and BHE, respectively (off the 8086's address bus) to tell the 8206 which byte is being written. The 8206 performs error correction on the entire word to be modified, but 3-states its DO/WDI pins for the byte to be written; this byte is provided from the data bus by enabling the corresponding 8286 transceiver. The 8206 then generates check bits for the new word.

During a read cycle, $\overline{BM_0}$ and $\overline{BM_1}$ are forced inactive (i.e., the 8206 outputs both bytes even if 8086 is only reading one). This is done since all cycles are implemented as read-modify-writes, so both bytes of data (plus check bits) must be present at the RAM data in pins to be rewritten during the second phase of the read-modify-write cycle. Only those bytes actually being read by the 8086 are driven on the data bus by enabling the corresponding 8286 transceiver.

The 8286's Output Enables ($\overline{OEB_0}$, $\overline{OEB_1}$) are qualified by the 8086's $\overline{RD}$, $\overline{WR}$ commands and the 8203's $\overline{CS}$ command. This serves two purposes: It prevents data bus contention during read cycles and it prevents contention between the transceivers and the 8206 DO pins at the beginning of a write cycle.

Thanks to the use of a 68-pin leadless chip carrier, the 8206 error detection and correction unit is able to implement an architecture with separate 16-pin input and output buses. Thus single-bit error correction may be added to a system with a minimum of control signals or external logic.□